Miracle3D: Team Description Paper for

RoboCup 3D Soccer Simulation Leagu

HongDao Xie，JianLei Jia，LanQi Ge，JiaHao Chen，Yang Le,Fan Wang

Hefei Normal University China

Abstract: This article introduces Miracle3D team and its development. It also describes the changes in Miracle3D team structure.In addit--ion, it also includes the research on deep learning optimization strategy decision-making.

1 Introduction

Miracle3D is a RoboCup 3D team founded in 2012 and has participated in many competitions. Miracle 3D simulation robot soccer team participated in the national competition for the first time in 2012. In 2013, Miracle3D won the Anhui Robot World Cup. In the same year, he won the first prize of Roboup3D China National Award and entered the top eight in Roboup3D IranOpen2014. In 2014, Miracle3D won the bronze medal of Robocup3D in Anhui, and won the fourth place in China in the same year.

There are still some problems with our team code. For example, the robot's strategy execution is inaccurate, the positioning is not accurate enough, the distance of kicking the ball is not far enough, and the movement speed is not fast enough. In order to accelerate the development of the team, focus on the research of problems.We uesd used the basic code released by MagmaOffenburg (if you wan to konw more about it,you can see https://github.com/magmaOffenburg/magmaRelease ）And we added our own strategy in the code.

The rest of this article is as follows. Part 2 introduces the basic code of MagmaOffenburg. The third part introduces our team structure. The fourth part introduces the design of neural network and the use of reinforcement learning to train robot player AI. Part 5 describes the future objectives.

2 Code Introduction

MagmaOffenburg RoboCup 3D simulation code is a highly modular code that makes our modification and development more flexible.

This code includes the following functions:

- Omnidirectional walking engine based on double inverted pendulum model
- A skill description language for specifying parameterized skills/behaviors
- Getup (recovering after having fallen over) behaviors for all robot types
- A couple basic skills for kicking one of which uses inverse kinematics
- Sample demo dribble and kick behaviors for scoring a goal
- World model and particle filter for localization
- Kalman filter for tracking objects
- All necessary parsing code for sending/receiving messages from/to the server

* Code for drawing objects in the RoboViz [4] monitor

* Communication system previously provided by drop-in player challenge 4

* An example behavior/task for optimizing a kick

3 Team Structure

The bottom layer includes communication module and receive/execute module. As the last layer of the layer structure, it is used to communicate with the server. Its functions include two aspects: sending and receiving. As the receiver, the communication module needs to obtain information from the server and send the message analysis model to the outside through the server. As the sender, the robot feeds back the decision to the server communication module in the following way, and then transmits the parsed information to the world model, and updates the world model according to the communication information.

The skill layer is also called the basic action layer, which defines the basic actions and skills, such as walking, shooting, positioning, interception, etc. The skill level is the foundation of the whole decision-making level and the bridge between the bottom level and the decision-making level. However, both the analysis of information and visual positioning will have certain deviation, which will affect decision-making. Relevant algorithms need to be used to reduce the impact. "Decision-maker" is the brain of the person who is responsible for coordinating the team strategy and making different positions, passes, dribbles, etc. according to the situation of the game.

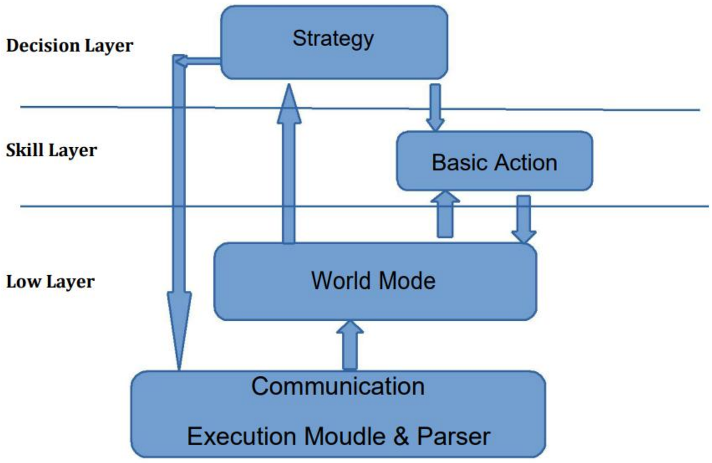

Fig. 1. Team Architecture of Miracle3D

4 Use reinforcement learning to train robot player AI

Using reinforcement learning to train robot player AI Preface: In order to improve training efficiency and reduce training time, first abstract robocup3d as a 2d small game, and use 2d small game simulation, which is similar to the traditional strategy in strategic decision-making. The traditional strategy also abstracts the field into a plane, and each player has a fixed state information representation.

4.1 Neural Network Design

Neural network, also known as artificial neural network (ANN) or simulated neural network (SNN), is a subset of machine learning and the core of deep learning algorithm. Artificial neural network (ANN) is composed of node layer, including an input layer, one or more hidden layers and an output layer. Each node is also called an artificial neuron. They are connected to another node and have relevant weights and thresholds. If the output of any single node is higher than the specified threshold, the node will be activated and send the data to the next layer of the network. Otherwise, the data will not be transferred to the next layer of the network.

After the player's position information and the ball's position information are input into the input layer, the entity is embedded into the position coding layer, the information is encoded, and then normalized through the residual connection and the self-attention layer. The global information is extracted through the average pooling, and the past strategy is memorized after entering an lstm layer.

LSTM, the full name of Long Short Term Memory, is a special recurrent neural network. This network is different from the general feedforward neural network. LSTM can use time series to analyze the input; It is used to solve the common long-term dependence problem in general recurrent neural networks. Using LSTM can effectively transfer and express the information in a long time series without causing the useful information in a long time to be ignored (forgotten). At the same time, LSTM can also solve the problem of gradient disappearance/explosion in RNN. It adds a memory cell, which is mainly composed of forgetting gate, input gate and output gate, as shown in the figure. The forgetting gate is used to control how much information in the memory cell of the previous moment is discarded or retained, the input gate is used to control how much information can be input and saved in the memory cell at the current moment, and the output gate is used to control which information in the memory cell will be output at the current moment.

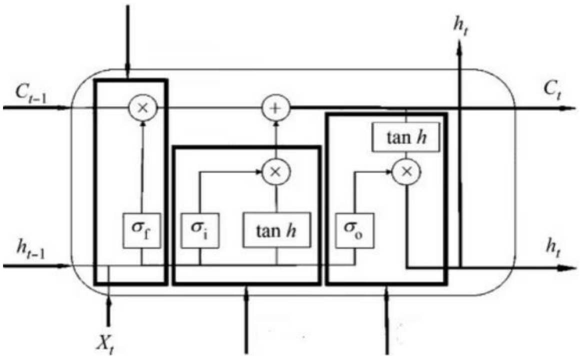

Fig.2 LSTM Structure diagram

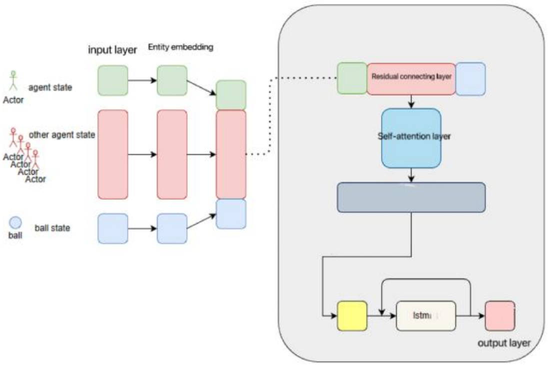

Fig.3 RL neural network model

4.2 Training Graphic Design

We have a 20 * 30m court with two goals in the middle of the left and right sides. Its length is 3m.

The ball is generated in the center.

The maximum number of players on both sides is 11, and the number can be adjusted at any time during training.

We will train a single agent first.

End condition: the ball collides with the goal

Robot player action:

1.Forward

2.Turn left

3.Turn right

4. Kick the ball (only when the ball is at a certain distance in front of the player)

5. Stand

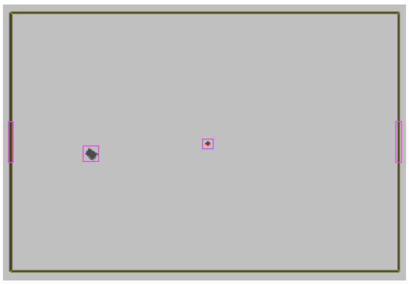

Fig.4 Plane picture of simulation training

4.3 Technical Implementation

4.3.1 Reinforcement Learning Algorithm

Reinforcement learning is an important branch of machine learning and a product of multi-disciplinary and multi-disciplinary intersection. Its essence is to solve the decision making problem, that is, to make decisions automatically and make continuous decisions.

It mainly contains four elements: agent, environment status, action, reward, and reinforcement learning aims to obtain the most cumulative rewards.

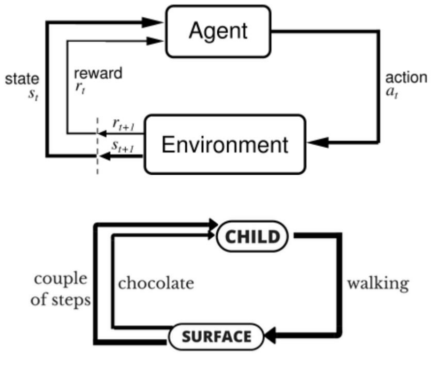

Fig.5 Process picture of reinforcement learning

4.3.2 PPO Algorithm

PPO algorithm is a policy-based reinforcement learning algorithm using two neural networks. By inputting the current "state" of the "intelligent body" into the neural network, the corresponding "action" and "reward" will be finally obtained, and then the state of the "intelligent body" will be updated according to the "action". According to the objective function containing "reward" and "action", the weight parameters in the neural network will be updated by using the gradient rise, so that the "action" judgment that makes the overall reward value greater can be obtained.

4.4 Reinforcement Learning Process

In the simulation plane we set up, we analyze the strategy execution steps of a single agent:

policy——action——>>env——state,reward——>>policy

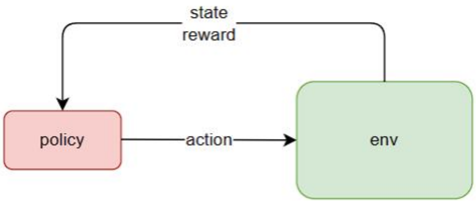

Fig.6 Training process

4.5 Result

The result is the stage reward analysis:

There is only one player below,

Set the feedback of feedback to encourage the agent to approach the ball. After about 500k simulations, the agent has learned how to go straight forward to approach the ball and then rotate in place.

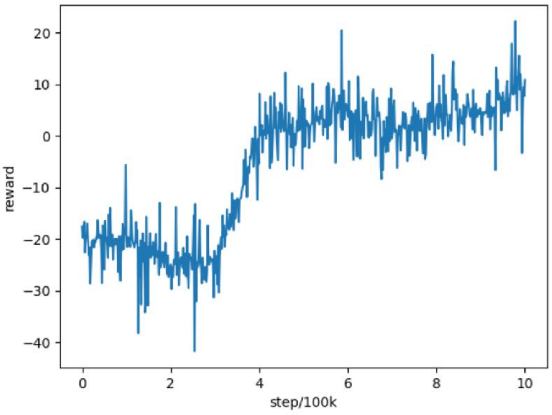

Fig.7 Training result

Adjust the feedback of feedback to encourage the agent to play football in the right position. The robot soon learned how to play football, but could not play in the precise direction.

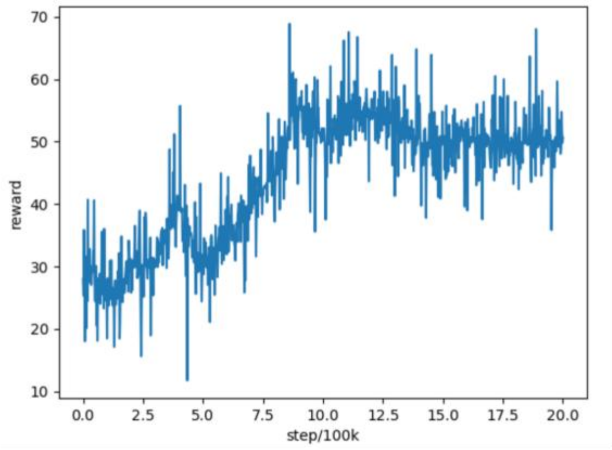

Fig.8 Training result

After adding a new opponent:

We set up a simple 1v1 confrontation scenario. The defender performs a fixed operation, walks to the side of the football, and then kicks the football. After about 750k simulations, the agent learns to kick the ball away from the opponent. After about 1.7m times, the agent learns to shoot the ball at the goal in two steps - first kick the ball away from the opponent, and then catch up with the football to finish shooting.

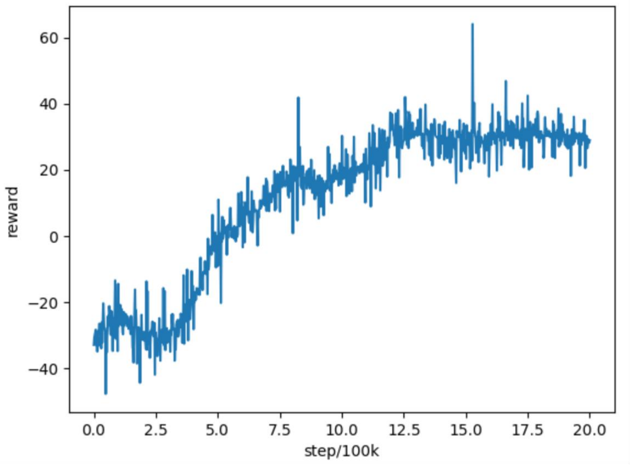

Fig.9 Training result

5 Future outlook

In order to solve the problem of strategy decision of soccer team members, it is necessary to repeatedly change the implementation of the code, study the decision-making methods of the team members, and establish a systematic robocup3d robot strategy decision mechanism. Based on reinforcement learning, relevant algorithms are used to train and optimize the agent's policy decision and parameter range. In the future, it will gradually improve the parameters of its training plane and support more players until 11.

6 Acknowledgements

Thank you. Now, our team is based on the basic code released by MagmaOffenburg. Thank the members of MagmaOffenburg for publishing stable basic code. We thank them for their efforts, publications and theories.

References

1 Patrick MacAlpine, Peter Stone. UT Austin Villa RoboCup 3D Simulation Base Code Release. In:Proceedings of the RoboCup International Symposium 2016 (RoboCup 2016),Leipzig, Germany, July 2016.

2 MacAlpine, P., Barrett, S., Urieli, D., Vu, V., Stone, P.: Design and optimization of an omnidirectional humanoid walk: A winning approach at the RoboCup 2011 3D simulation competition. In: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI-12) (July 2012)

3 MacAlpine, P., Urieli, D., Barrett, S., Kalyanakrishnan, S., Barrera, F., Lopez-Mobilia, A., Ştiurcă, N., Vu, V., Stone, P.: UT Austin Villa 2011: A champion robot inthe RoboCup 3D soccer simulation competition. In: Proc. of 11th Int. Conf. on Autonomous robots and Multiagent Systems (AAMAS 2012) (June 2012)

2 MacAlpine, P., Barrett, S., Urieli, D., Vu, V., Stone, P.: Design and optimization of an omnidirectional humanoid walk: A winning approach at the RoboCup 2011 3D simulation competition. In: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI-12) (July 2012)

3 MacAlpine, P., Urieli, D., Barrett, S., Kalyanakrishnan, S., Barrera, F., Lopez-Mobilia, A., Ştiurcă, N., Vu, V., Stone, P.: UT Austin Villa 2011: A champion robot inthe RoboCup 3D soccer simulation competition. In: Proc. of 11th Int. Conf. on Autonomous robots and Multiagent Systems (AAMAS 2012) (June 2012)

4 O’Rourke J． Computational Geometry in C［M］． Beijing: Mechanics 5. Industry Press ,2005: 161— 165．

5 Bowyer A． Computing Dirichlet tessellations［J］.The Computer Journal,1981,24(2) : 162—166．

6 Amenta N,Bern M,Kamvysselis M． A new Voronoi —basedsurface reconstruction algorithm ［C］/ / Proceeding of SIGGRAPH’98． Danvers: Assison—Wssley Publishing Company,1992: 415—421．

7 Edelsbrunner H,Shah N R． Incremental topological flipping works for regular triangulations［J］.Algorithmica,1996,15(3) : 223—241．

8 Rongyi He,Chunguang Li. RoboCup3D simulation robot gait optimization research [J]. Computer and Modernization, 2018 (3).